Practical evaluation of SEEK and OpenBIS for biological data management in SynthSys; first report.

Practical evaluation of SEEK and OpenBIS for biological data management in SynthSys; first report.

Eilidh Troup[2], Ivan Clark[1], Peter Swain[1], Andrew J. Millar[1] and Tomasz Zielinski[1]

30 October 2015

[1] SynthSys and School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3BF, UK

[2] EPCC, University of Edinburgh, Edinburgh EH9 3FD, UK

Author contributions: ET and TZ evaluated systems and developed software; IC, PS and AJM provided use cases; AJM and TZ designed the evaluation; ET, TZ and AJM wrote the report with input from all authors.

Funding: The work described here was supported by the BBSRC and EPSRC through SynthSys-Mammalian, award BB/M018040/1; early scoping was supported by EU FP7 project TiMet (award 245143).

Acknowledgements: We gratefully acknowledge training and support from the SEEK and OpenBIS project teams, who also checked this document.

Contents

Executive Summary

Objective

The project evaluated two existing data management systems for a small set of users, who represent diverse needs within the SynthSys Centre, in order to inform wider adoption for biological research.

Background

Modern data intensive research requires systems to process and organise electronic data for collaborative work and for public dissemination. Moreover, research funders are requiring that their scientific output is made publicly available. With this project, SynthSys seeks to address our shared and pressing need for user-friendly, data management systems.

We identified two management systems for biological data: SEEK and OpenBIS.

We aimed to find a product with additional benefits that engage and motivate users, over and above meeting open data requirements. The data deposition should be easy for biologists and automated where possible. The data should be easy to search and browse and the systems should be extendable to allow customised, specialist data analysis or visualisation.

Approach

We performed an intense, practical evaluation of two systems by implementing selected use-cases. The first use-case (PS) was from Peter Swain's lab. In a typical experiment, a plate reader monitors the optical density and fluorescence of yeast cultures and the collected timeseries data are then used to calculate growth rate and gene expression for different yeast strains. The second use-case was replication of PlaSMo, an existing web-repository for plant systems biology and growth models. PlaSMo stores models and associated assets such as supporting data or images.

Results

We were able to fully cater for the PS use-case using OpenBIS and extensions developed by us. Building on top of the existing OpenBIS API, we implemented automated metadata extraction and triggered custom data processing. In SEEK, we also developed automatic metadata extraction and provided custom search. However, SEEK lacks an API for integrating data analysis.

Members of the Swain lab preferred the OpenBIS solution due to the automatic data processing.

SEEK replicates the PlaSMo functionality very well, capturing almost the same metadata and allowing similar control over data sharing. On the contrary, we concluded that OpenBIS was unsuitable on account of its restricted permission model.

Outcome

SEEK’s strengths are support for the Investigation, Study, Assay (ISA) standard and a fine grained access control. This makes SEEK an excellent tool for collaborative work and publishing results. OpenBIS is well suited for automatic metadata processing and incorporation into analysis workflows.

Both data management systems provided useful and complementary functionality, so our recommendation is that both are hosted for use in SynthSys. This also aligns well with the EU FAIRDOM project which is currently integrating SEEK and OpenBIS into one platform.

Motivation

21st century science is governed by *Data-Intensive Scientific Discovery*, which data bridges the other three scientific paradigms: theory, experimentation and simulation. For the multidisciplinary, collaborative research within the SynthSys consortium, data plays a critical role. Moreover, funding bodies acknowledge the importance of data and require a data management strategy for funded projects. As a result, data management has become an important part of modern research and software infrastructure is necessary to support the whole data life cycle, from data acquisition, through analysis, to data sharing.

However, individual research groups may lack expertise and resources to setup data management solutions themselves. Technical aspects like provision of servers, backup solutions, system administration and integration with the University's infrastructure (authorization mechanisms, DataShare) are more fitting on a centre- or School-wide level. Besides, laboratories face similar concerns: creation of inventories for biological materials (seed stocks, cell lines), tracking relationship between primary and secondary data, linking to publications and external resources, organizing data into logical structure (research projects). Providing data management as a service within the Centre would reduce the maintenance costs, improve research practice and facilitate knowledge exchange. An ideal data management system becomes part of the workbench and assists in the research by solving problems or providing "extra value" to the user, for example, in the form of data processing or visualisation.

Here, we evaluate two existing open-source systems, SEEK (Fairdom version 0.24; www.seek4science.org) and OpenBIS (official release 13.04 and Sprint release 214; www.cisd.ethz.ch/software/openBIS). Each has a broad existing user base: SEEK >50 institutions in Europe; OpenBIS >20 institutions, mostly in Switzerland. This is the first report of an ongoing, practical evaluation by implementation of real use cases that represent diverse needs within the Centre, in order to assess more general usage in biological research and the costs involved.

Features of DM systems under evaluation:

- Web-based user interface
- Security model that allows both privacy and data sharing
- Organization of data into a logical structure that helps with navigation and data browsing
- Metadata model, its flexibility, customization and support for closed vocabularies or ontologies
- Data description process
- Relationship between data, e.g. primary-secondary data
- Search options
- Programmatic access to data and metadata, API for integration with data processing or visualization

Use case description

We selected two use cases, one based on numerical data generated from biological experiments, the second one focused on theoretical aspects of modern research.

Peter Swain Use Case (PS use-case)

Peter Swain's group (http://swainlab.bio.ed.ac.uk) provided us with our first use case. They study cellular decision-making in response to nutrients using a fluorescence plate reader. In a typical experiment, individual wells of a 96-well plate are filled with cultures of different yeast strains, suspended in media that vary in sugar content. The yeast strains express a Green Fluorescent Protein (GFP), so that the expression levels of certain genes can be measured. The plate reader measures optical density and fluorescence at selected wavelengths over time (typically 24 hours). The optical density (OD) is used to measure the number of cells and the fluorescence is used to measure gene expression. The gathered data are then analysed using a custom python script, which for each timeseries corrects for autofluorescence and also uses the OD measurements to calculate the average level of fluorescence (gene expression) per cell. Graphs of these results, illustrating growth and gene expression, are generated for each combination of yeast strains and sugars under the study (See Figure 1d) Using this data the researchers can draw conclusions about how cells decide to utilise different nutrients in complex and changing environments.

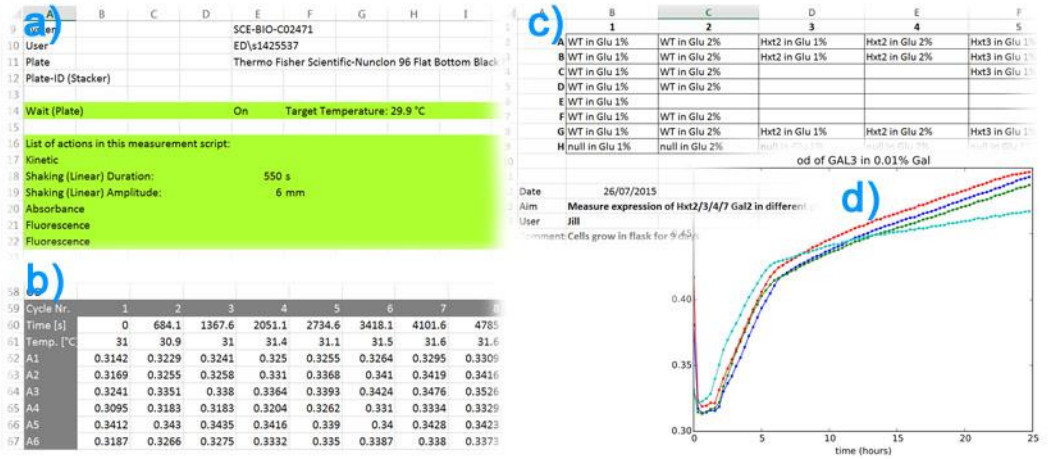

Figure 1. PS Use-Case Data: a) sample of technical parameters in plate reader file; b) timeseries data in plate reader file; c) plate content description; d) sample of analysis results

The plate reader produces an Excel file containing the measured absorbance/emission levels at each timepoint, accompanied by technical parameters of the measurement, which are written by the equipment itself (see Figure 1a,b). In order to analyse the data, the strain and sugar content of each well must be known. Currently this information is represented in another Excel file as a table which mimics the 96-well plate layout, in which each cell contains name of the strain used and medium composition (see Figure 1c). This file contains some additional metadata like the aim of the experiment, the date and its authors.

PS group was interested in a solution that:

- provides a consistent way of organizing their raw data and the results
- allows retrieval of data for a particular yeast strain and sugar
- preserves existing technical metadata
- can automate the data analysis
- can display timeseries data and fit.

Plasmo Use Case

The second uses case was reproducing Plasmo functionality. Plasmo is a repository of models for Plant Systems biology (www.plasmo.ed.ac.uk). A typical Plasmo record contains (Figure 2):

- a general description of the model,
- the actual model file in one of the supported XML formats (SBML, SIMILE, SBGN...),
- images illustrating the model structure,
- supporting data files
- linked literature resources

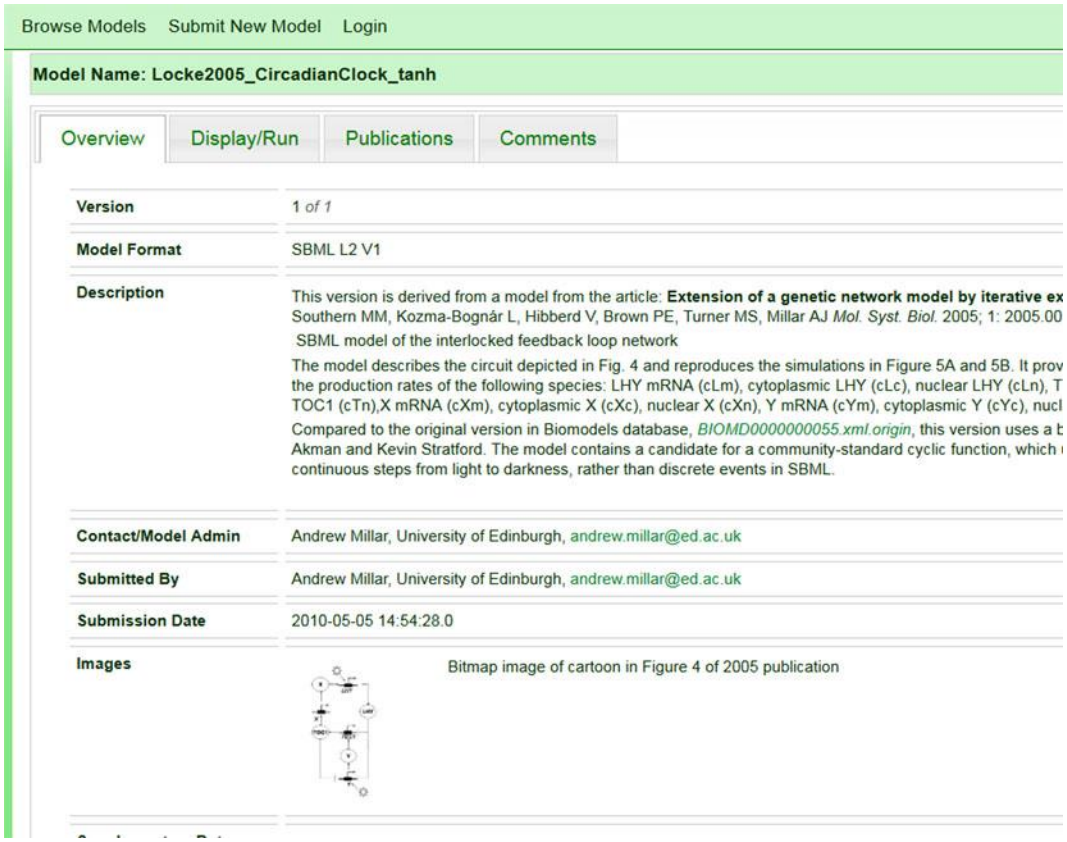

Figure 2 Example of model record in Plasmo.

Plasmo offers additional features apart from simply storing files in organized manner. It offers free text search as well as a browse option. The model files are validated upon upload to assert if they complied with the selected format. For some model types, Plasmo can list the model components by reading the XML, visualise the model or even run the model using an external service. It also offers the option of adding comments to the model. Plasmo supports versioning of the models, in the sense that consecutive evolutions of the model can be stored under the same “global” model identifier and then referenced by version number, each having an independent description and potentially different supporting files (inheritance of supporting files from the parent model is also supported). Plasmo's security model allows sharing a model only with a selected group of users, which allows collaborative work on the model. Finally each model can be uniquely identified by its permanent URL address.

We were interested in mimicking the majority of the above Plasmo features.

Implementation of the use cases in SEEK

Peter Swain Use Case

Organisation of raw data and results

The data for the Swain lab use case consists of a data file generated by the plate reader and a meta data file written by the user which describes the strains and sugars. The aim of the experiment is described in increasing detail in the investigation, study, assay hierarchy in SEEK. The data files are then loaded into SEEK and associated with an assay. Each file is loaded individually and it is the common assay that they are associated with that links them together.

Finding data for a particular yeast strain and sugar used

SEEK provides a simple search function. All of the words entered into the SEEK as titles, descriptions, etc. are indexed, as are the contents of data files such as the spreadsheets produced by the Swain Lab. The user is able to search for data files containing certain sugar or strain names, but it is free-text search so it misses the context in which those names appear and does not allow to search for a strain grown in a given sugar (i.e. both together in the same sample).

SEEK can parse RightField-compliant Excel spreadsheets into a knowledge graph which is stored in an RDF (Resource Description Framework) database (Virtuoso). RightField spreadsheets can have ontologies embedded within the cells of the spreadsheet, which produce a dropdown list of options for the user to select (Figure 3a). Due to this feature RightField templates can represent metadata in unambiguous way (Figure 3b). RDF is a powerful, widely-used technology that can facilitate automated reasoning. However, as we discovered, SEEK doesn't provide any way to search or browse the data that it stores into the RDF knowledge graph database.

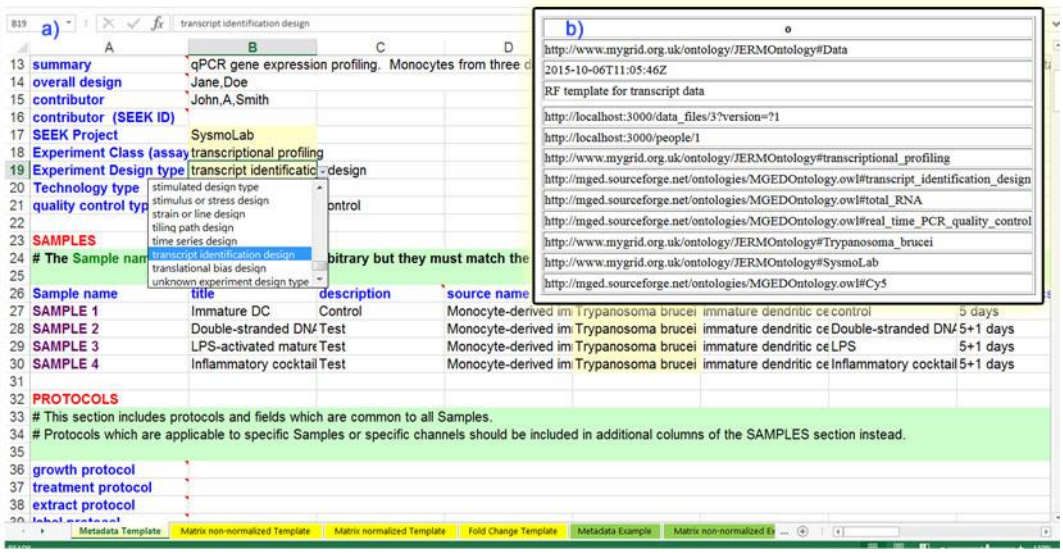

Figure 3. a) RightField template with dropdowns for ontology terms; b) the corresponding description in the RDF database

We considered creating a RightField template for users to enter the sugar and strain meta data into, but this had a couple of problems: over 100 strains in the dropdown list would be too many to use in practice. Three cells would be needed to represent content of one well: two dropdowns for the strain and sugar; and one cell for the concentration of the sugar. Creating 3 cells in the spreadsheet for every well would lead to 3*96 (288) values to be entered for every metadata file. This spreadsheet would be unwieldy to work with and difficult to read. Instead, we decided to work with the file format that is currently being used by the PS lab (Figure 1c).

We wrote new code in Ruby to parse the PS metadata file and save the extracted information into the RDF database. We captured details such as that a sample (well) was measured for a certain strain grown in a certain sugar. Our custom code was linked to existing SEEK code for file indexing upon data upload.

We also needed to add a new search page to SEEK for searching specifically for a strain, a sugar, or a combination of both in one well of a plate within experiment (Figure 4). The form invokes our custom query to the RDF database. The matching results are transformed into SEEK entities (Assay, Data File) and rendered in the page using the existing SEEK code.

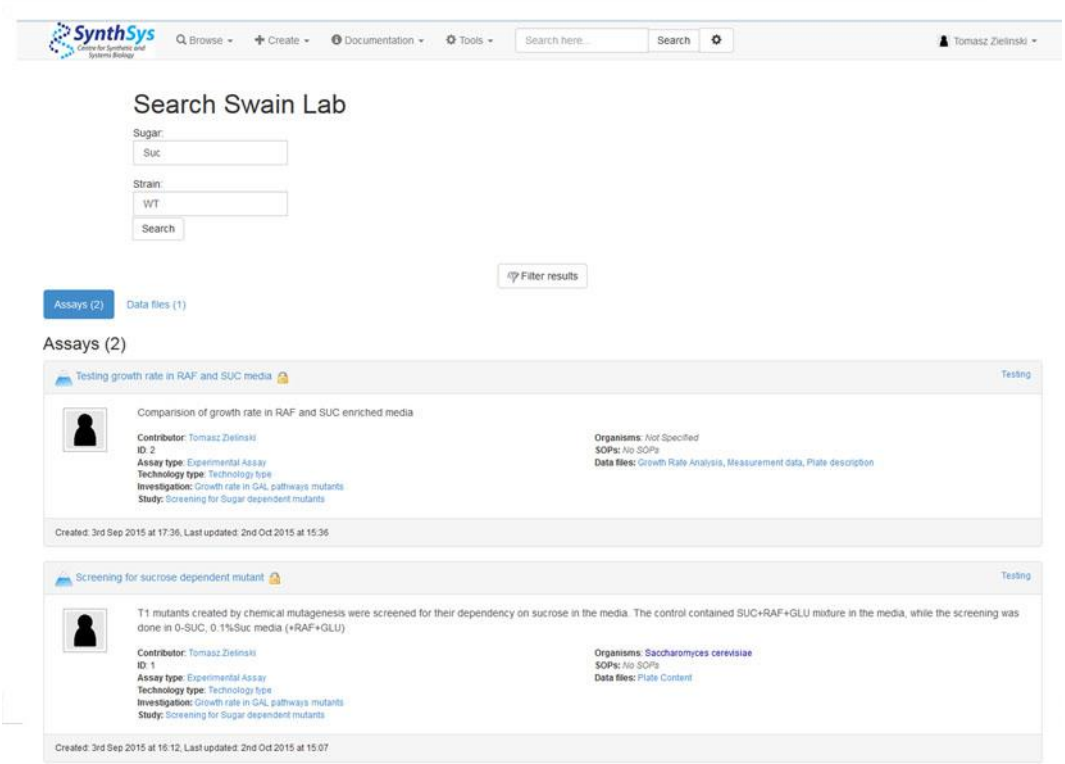

Figure 4. PS Custom search screen. The results list is rendered using built-in features of SEEK.

Leveraging the RDF database for indexing and searching has the advantage of being the most general and flexible solution. For example, for some use-cases RightField templates may be sufficient to capture metadata and we would only need to provide a custom search function, which should be easy to build by re-using our code. This approach can also benefit from future improvements to the SEEK, enriching RightField templates and new features utilising the knowledge graph.

This solution works but it reveals one of SEEK's shortcomings. Programmer effort is required to benefit from structured, detailed metadata; an administrator would not be able to do something similar for another data format.

Processing of data and display of results

There is no facility in SEEK for automatically running the python scripts that process the raw data and generate graphs. We considered having an external process running the analysis and generating the output files. However, there is no feature in SEEK that automatically detects data files as they are created by users, to trigger the analysis. It turned out that SEEK's data "harvesters", which were meant to do this kind of task, had been discontinued and we were discouraged from using them.

Once a user generates the graphs from the data they can be uploaded into and viewed in SEEK. Excel spreadsheets can be explored and annotated without the need to download.

Plasmo Use Case

SEEK has more of a hierarchical data structure than Plasmo. Plasmo represents a model with one model file and associated files and images. SEEK stores data under the ISATAB format (http://isacommons.sourceforge.net). This has a hierarchy of Investigation, Study, Assay. Assays can either be 'experimental assays' or 'modelling analyses'. Model files belong under 'modelling analysis' type assays in the hierarchy. More than one data file can be associated along with a model as part of the same assay.

Plasmo versioning of the models can be mimicked by creating an individual assay for each version of the model and storing them under a common Study (Figure 5). In that way each model version (Assay) can have its own description and collection of supporting data files.

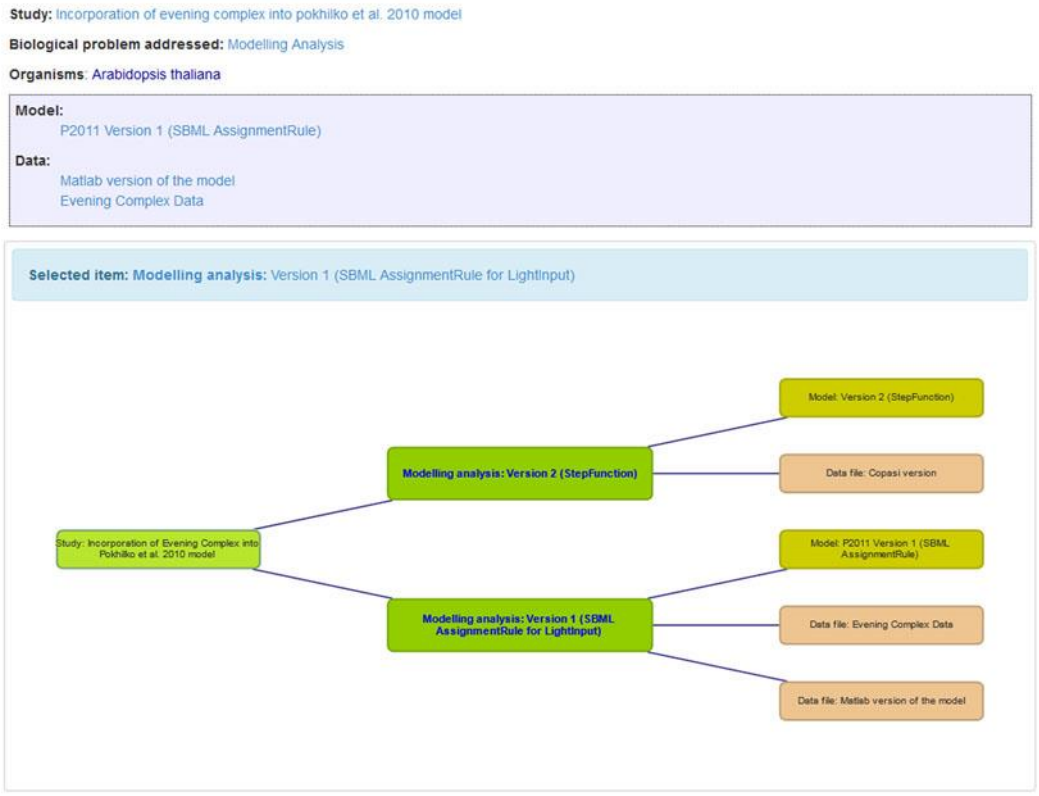

Figure 5. Representation of model repository in SEEK. In order to mimic the PLASMO versioning, two assays (versions) were created, each having its own description and related files

Feature mapping.

To investigate how well the data that is captured in PlaSMo could be stored in SEEK, we went through each option that is available in PlaSMo when uploading a new file and mapped it to the closest corresponding option in SEEK. The results are shown in the table below.

	PlaSMo	SEEK
Model Formats	SBGN-ML PD SBML L2 V1 SBML L2 V2 SBML L2 V3	SBML CellML SciLab XPP

	SBML L2 V4 SimilieXMLv3 XGMML1.0	BioPAX VCell Matlab package R package Mathematica MFAML
Model checking	Checks uploaded file conforms to specified format	Does not check format of uploaded file.
Model name	same	same
Model description	same	same
Access control	restricted to one or more groups - add to a project	Sharing – private; with members of this project; with members of any project; with all visitors. Various levels of permission (view summary, contents, edit permission). More options for sharing data are available…
User defined attributes	name, value pairs	No exact equivalent. Tags, possibly.
Images/screenshots	Multiple images + descriptions with model	One image file can be uploaded with model. If more images are needed or they need a description then they'd have to be uploaded as a data file and associated with the same Experimental assay or Modelling analysis as the model is.
Supplementary data files	Supplementary data files + descriptions	Uploaded as a data file and associated with the same Experimental assay or Modelling analysis as the model is.
References	Journal reference Textbook reference	Publications
Investigate/run model	Display contents of model (sub models, compartments, variables). Integrated with simileweb to browse/run simulation.	Can use "JWS online" service to run SBML simulations.

For most items there is a good mapping between PlaSMo and SEEK. However, PlaSMo allows a user to define new attributes in the form of name-value pairs. This provides flexibility and is not available in SEEK. PlaSMo checks that the format of uploaded files matches the specified format. SEEK supports this use case within a standard system, avoiding PlaSMo's separate software, web interface and database, at the cost of PlaSMo's separate identity and potentially some flexibility.

Implementation of the use cases in OpenBIS

Peter Swain Use Case

Organization of the data and results

The plate reader output file and the results of python analysis can be logically represented as the contents of the OpenBIS DataSet. A specific DataSet type was defined which is characterised with attributes matching the technical details recorded by the plate reader (e.g. measurement temperature, monitored wavelength etc.) (Figure 6). Each DataSet belongs to an Experiment, which has its own properties: its aim, authors and the most important names of sugars and strains used within experiment (Figure 7). Experiments are further organized into specific projects depending on the user needs (e.g. individual workspaces or drug studies).

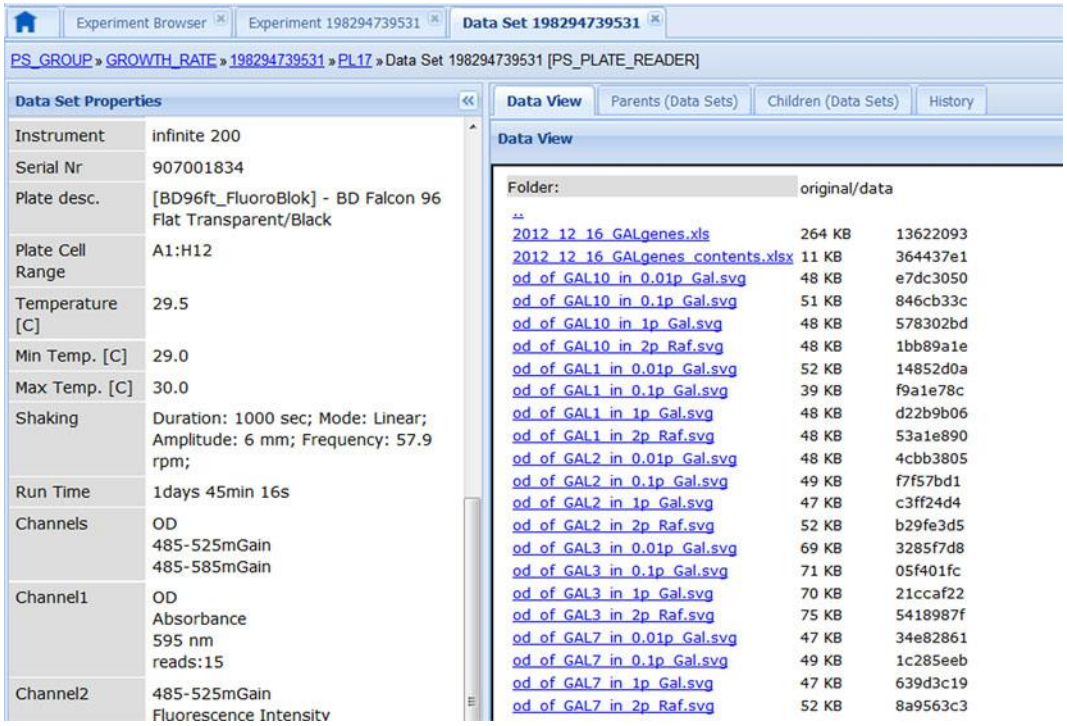

Figure 6. OpenBIS Data Set view

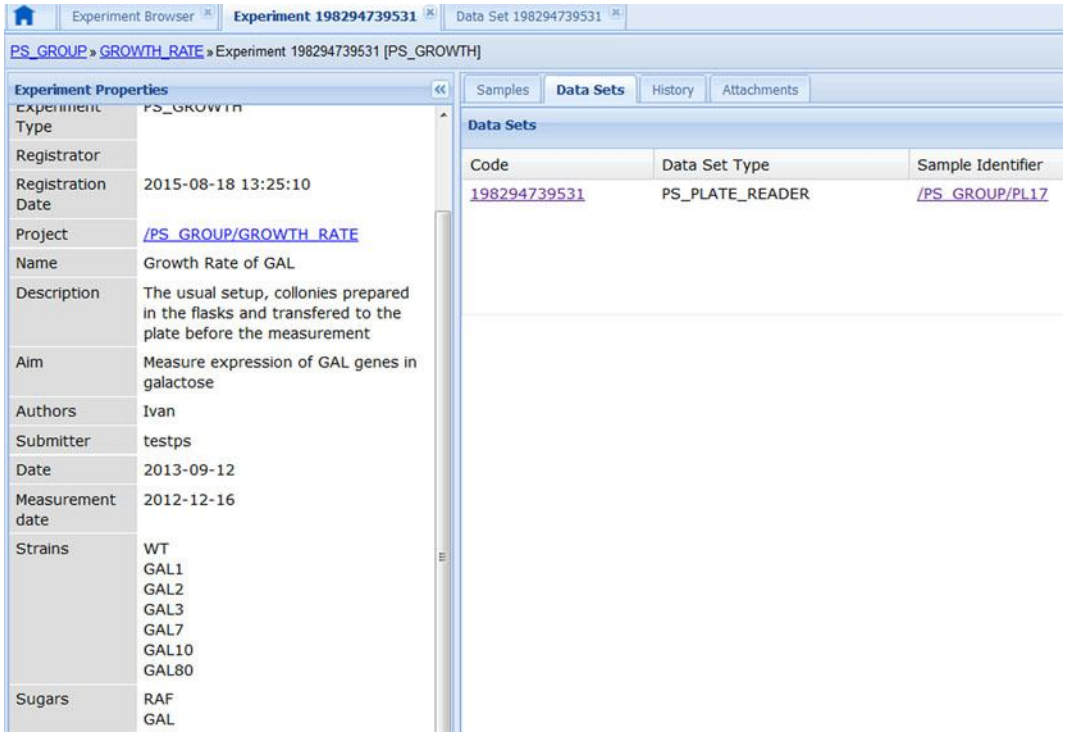

Figure 7. OpenBIS Experiment view

Data access and search

OpenBIS provides a tabular overview of the experiments grouped in a project (Figure 8). Table content can be sorted by individual properties of an experiment (table columns) and filtered by required attributes. This feature allows quick browsing and selection of interesting data sets.

OpenBIS also has a built in search mechanism. The user defines search criteria by selecting interesting attributes of Experiment/DataSets and their requested values. In this particular use case, the search is for data obtained using a given strain and sugar, with specific values for each (Figure 9).

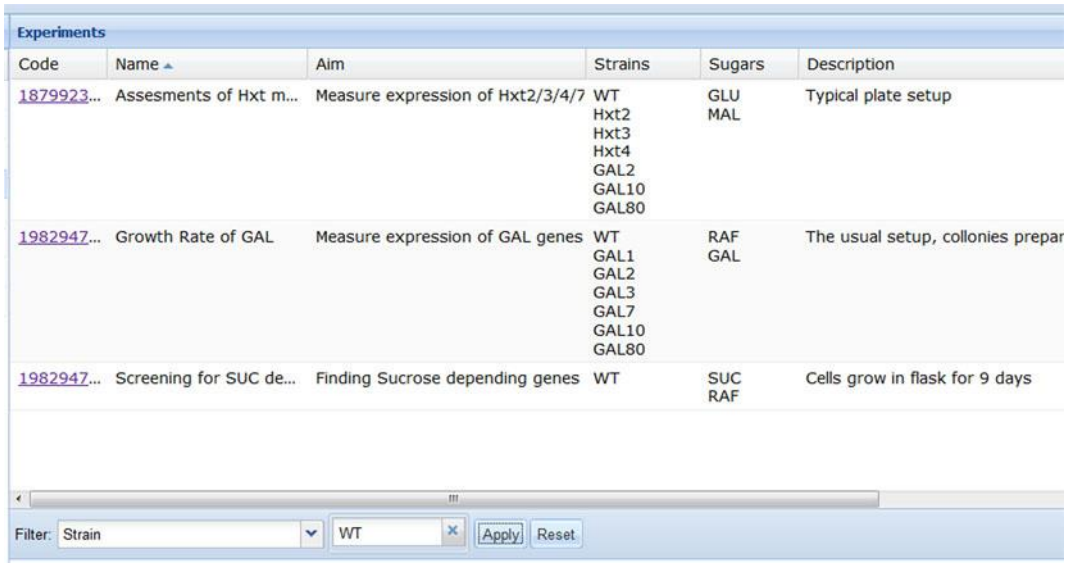

Figure 8. Tabular view of the experiments, sorted by their names and filtered using strain value

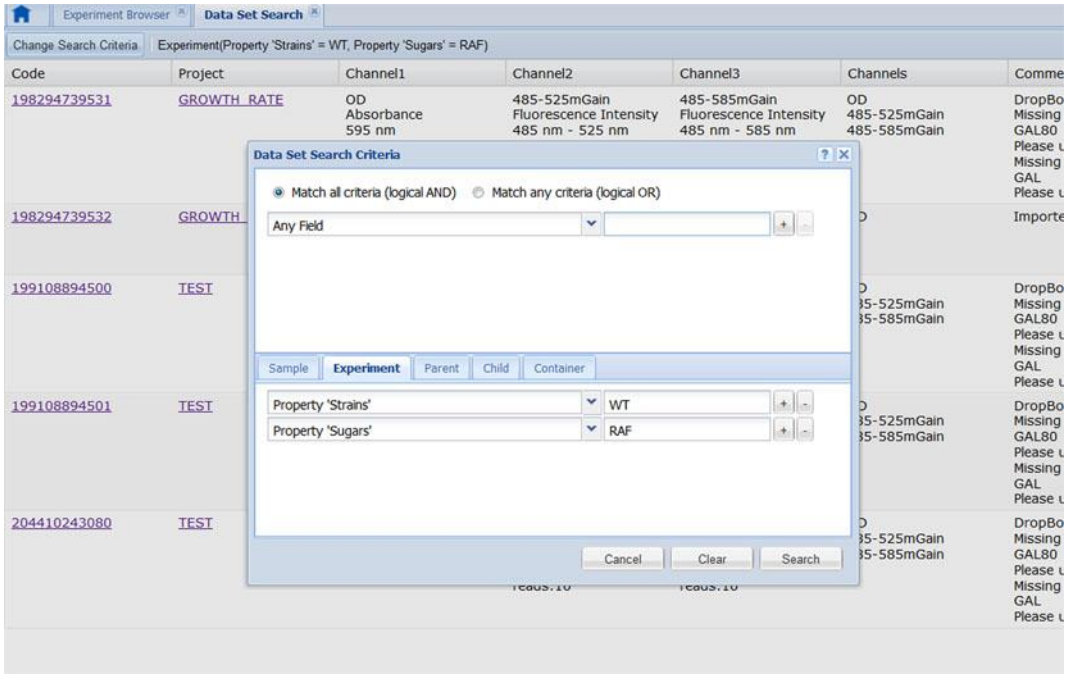

Figure 9. OpenBIS search form

Another useful way of accessing data is by exploiting the parent-child relationships that are possible in OpenBIS. We imported PS group strains and sugars inventories into OpenBIS by representing them as specific sample types with attributes matching their current database descriptions (Figure 10). Samples representing strains and sugars can be then selected as parents for particular experimental data. Thanks to this, all the experiments that used a particular strain can be listed by accessing the definition of that strain. Such linking between inventory entries and experiments assures rich information without unnecessary metadata redundancy.

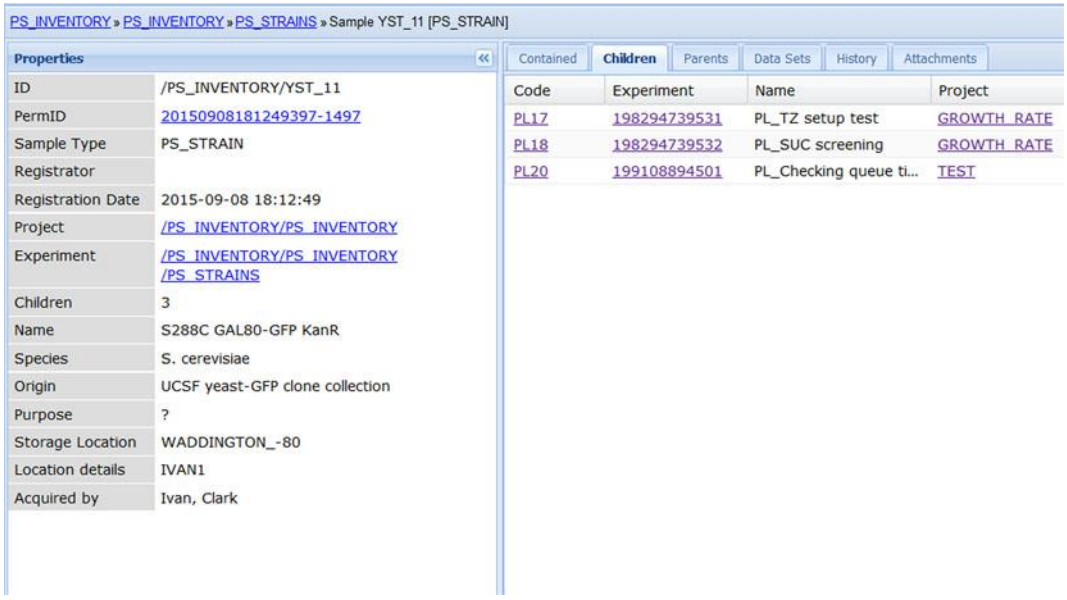

Figure 10. Strains inventory implemented as OpenBIS samples. Children tab list experiments that used the current strain

The parent-child relationship can be graphically visualised and capture deeper levels of hierarchy, like the dependency between strains and the plasmids used for their construction.

Once an interesting experiment is found, the content of its data set is presented as a list of files (Figure 6); any image files present can be displayed (Figure 11).

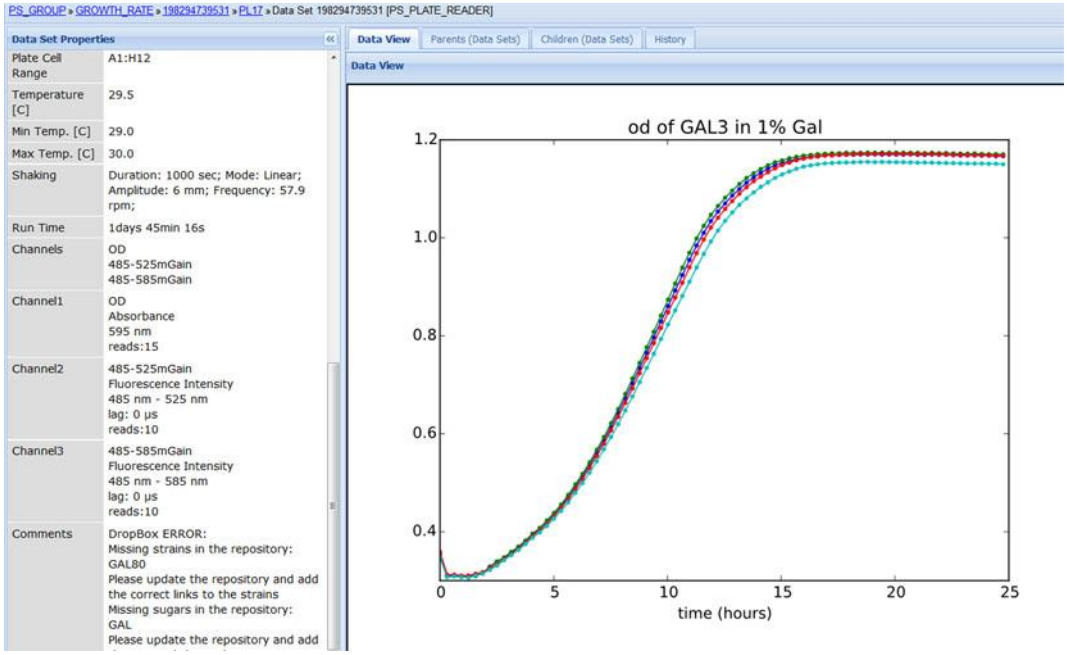

Figure 11. Data Set view with rendered graphical content.

Data description and upload

OpenBIS contains a UI for describing the values of attributes for Experiments/DataSets (Figure 12).

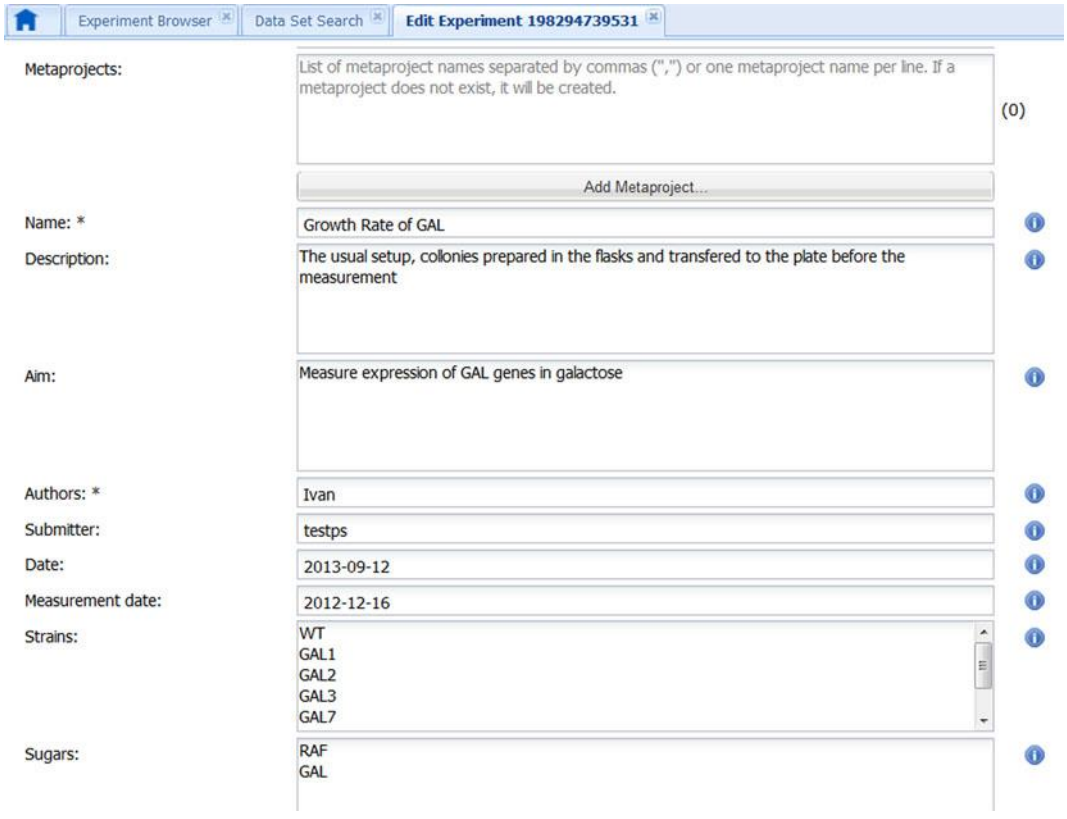

Figure 12. OpenBIS metadata entry and editing screen.

However, in our case, using the UI would be detrimental for the users, as all of the metadata are already present in the plate reader output and plate content description file.

Fortunately, OpenBIS provides automatic metadata extraction using a “dropbox” mechanism. Dropboxes are folders on the servers that are monitored by OpenBIS. Upon discovery of new content in the dropbox, a Jython script is invoked which can process the new entries and import them into OpenBIS. We developed utility software in Java that extracts technical and experimental metadata from files uploaded in the PS lab format, which are used in the next step by a Jython script to create fully annotated DataSet and Experiment entries in OpenBIS. The script also links the newly created entries to appropriate entries in the Strains and Sugars inventories. Creation of these new records in OpenBIS is fully automated and does not demand additional work from the user.

Nevertheless, direct access to dropboxes by the users was not a convenient solution. Firstly, dropbox folders are located on the server, to which access should be limited for security reasons. Secondly and more importantly there is no feedback mechanism to the users. If the upload fails, it manifests itself only by the lack of a new data entry in the OpenBIS, which not only means that users need to monitor OpenBIS content but also that they will be oblivious to the reason for the failure. For example, in Peter Swain’s use case, two different files (machine output and user description) are necessary to create a new data entry. Omission of one of them or submission of a file in the wrong format would crash the upload process.

For that reason, we developed a simple web application that is responsible for placing data files in the OpenBIS dropbox. The web form allows the user to select experiment type, relevant data files and the existing project into which they should be added (Figure 13). This web application reuses the same Java code to validate the submission as the dropbox script for metadata extraction. If some format constraints are not met it provides instant feedback to the user. Then it copies the files to the appropriate dropbox folder and monitors OpenBIS to discover if upload was successful, presenting the user with progress status.

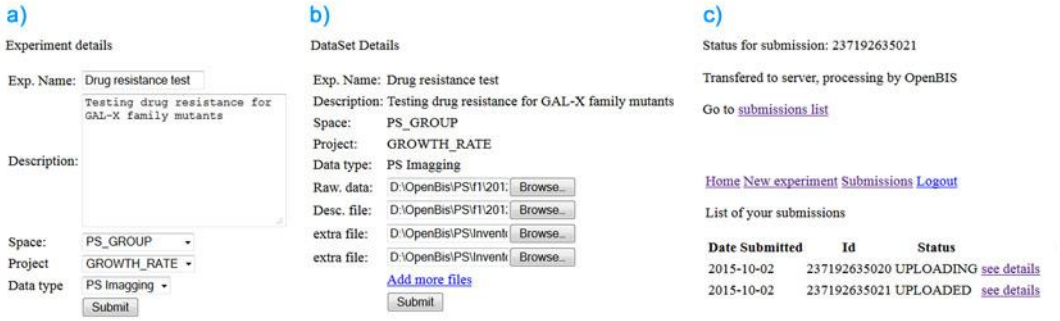

Figure 13. Web application for data uploads to OpenBIS. a) Experiment description and b) data selection c) Upload status screens.

Automation of the data analysis

Initially, we thought of using dropbox scripts in OpenBIS to invoke the automated data analysis required by the PS lab. However, it was not a technically sound solution. The data processing may be a computationally extensive task which could potentially fail. Invoking a time consuming process directly from OpenBIS would hold up other data uploads. If the processing hangs due to an error it would block the whole system. The dropbox scripts were meant to perform simple metadata pre-processing and their use should be limited to such.

Instead, we extended our web application for data upload with processing capabilities. It can be configured in such a way that for defined experiment types an external program/script is invoked to analyse the data. The external analysis has to save its results next to its input data and upon its completion the whole pack of raw and generated files is moved to the correct dropbox. Once the files are in the dropbox, the previously described process of metadata extraction and inventory linking takes place.

This is a general and flexible solution:

- custom analysis procedures can be easily added
- the queuing mechanism for data processing utilizes multithreading so different uploads can be processed concurrently, using whatever processors are made available
- it provides feedback to the users in case of processing errors

To summarise, we were able to meet all the objectives of the PS use case by combining features of OpenBIS with our own web application for data upload. As for SEEK, the software development required for the PS use case could not be accomplished by an administrator.

PlaSMo Use Case

After initial evaluation we did not pursue the implementation of this use case in OpenBIS.

The deciding factor was the constrained security model of OpenBIS, which limits collaboration options and does not allow granting public access to already existing resources. Also model descriptions do not have the rich structure which would benefit the most from the OpenBIS metadata handling. Finally, out of the box, OpenBIS does not provide support for publications or external resources as SEEK does. In light of the fact that SEEK provided the majority of PlaSMo functions, we decided not to commit more effort into the OpenBIS implementation.

Strengths and limitations of the two systems

Seek	OpenBIS
Security model: data access and service management	
Strengths: Very flexible: • access can be defined for individual data files, studies, assay etc. • access can be granted from coarse (whole institution) to fine level (named users) • allows full or summary only view of the entries	Strengths: Authorisation can be configured to use custom mechanism e.g. LDAP
Limitations:	Limitations:
• authorisation only via Seek mechanism • projects (equivalent of user groups) can be created only at the administrator level	Very constrained, simplified model: • access is defined on the space level (container for projects/experiments) and

	cannot be decided for individual experiments • resource visibility cannot be modified once an entry is created. It is determined by the containing space and the assignment to a space cannot be changed using the UI Metadata management limited to administrator level: • definition of Experiment/DataSet types • content of vocabularies lists
Meta data	
Strengths:	Strengths:
Support for RightField documents that allows metadata description using templates annotated with ontology terms and selection of possible values. Templates are created by expert users/data managers, without programming. Representation of metadata as a knowledge graph in an RDF (Resource Description Framework) database (Virtuoso). User defined tags for simple annotations. There are links between data files and assays, and these fit into the standard Investigation-Study-Assay hierarchy.	Metadata are represented as a set of properties (parameter/value pairs), which are defined individually for different experiment/DataSet/sample types. Properties can be validated or handled by custom code Support for closed vocabularies and materials (special terms in a vocabulary that are not just labels, but have their own set of properties) Parent-child relationships between entries
Limitations:	Limitations:
RightField template does not support relations between fields: for example it is not possible to express that the factor studied was a temperature of value 37C, only that temperature was the factor that was studied. The metadata contained in RightField documents are not propagated to the parent SEEK entities like Study/Assay, for example there is no automatic linking between Assay/User and data file nor Assay description, nor extraction of factors studied from the data file. The RDF knowledge graph is actually not being used in SEEK. It is not being used for querying nor used for reasoning	Metadata types defined on admin level, not editable by users (experiment/dataset types, vocabularies) All the data/experiment/properties types and their properties are visible to every user, potentially cluttering up the UI if used for a School or Centre, where each lab defines multiple, different types No relationships/constraints between experiment, dataset, sample types, i.e. experiment of type PCR can contain data of type Microscope Imaging. Lack of ad-hoc properties or annotations
Logical structure	
4 level structure Investigation -> Study(ies) -> Assay(s) -> Data Files is flexible enough to	Project -> Experiment(s) -> DataSet(s) -> DataFiles is flexible enough to organize research data. In reality the structure contains also

organize research data and capture relationships between them. Additional hierarchical structures are also implicit in SEEK: (Funding) Programme -> Project Event -> Presentation -> File Institution -> People -> Profile Strains, Cultures and Samples are currently being re-programmed by the SEEK team; current properties are not discussed here. Entities (Study, Assay) under one Investigation cannot be shared with another, for example if results from one Investigation become the starting point for another. Data and Model files can be linked to multiple Assays, including from different Investigations. The Assay metadata must be entered separately.	Sample(s) between Experiment and DataSet but those are used to represent biological/technical details rather than to organize data. Samples are also a convenient way of representing Inventories entries (Chemical, seed stocks, etc) as they are more flexible than Material type which was originally designed for that reason
Data browsing	
All SEEK entities (Investigation, Study, Models etc) can be browsed using a list view with summary information for each item. Browsing can be limited to the latest items, or to items with names starting with selected letter. Depending on the entity category it can be further filtered, for example by tags or assay type. Once an entity is selected a navigable, visual representation of its relationship with other items (containing studies, related publications etc) is available.	Experiments and Samples can be browsed and the browsing can be limited to given experiment or sample types. The tabular view of the items displays all their properties, individual properties can be used to further filter the results. Once item is selected there are tabs allowing access to related datasets, experiments, samples or parent/children elements.
Searching	
Only free text search is available, based on both items descriptions and indexed file content. Results are grouped by categories (assays, models, etc) and can be further filtered by tags (if added by users) or type specific elements (e.g. model format type).	Free text search is available; it uses the metadata information but no actual file content. Search can be limited to a given entity category (experiment, DataSet). There is also a powerful, complex DataSet search in which specific criteria can be defined using attributes of the DataSets as well as properties of related to it experiments and samples. For example it is possible to search for DataSets belonging to experiments authored by a specific user and related to biological samples having particular genotype.
Data description and deposition	

There is an intuitive UI for description of the entities and defining relationships between them. DataFiles, models or publications can be directly uploaded or linked to external repositories. The metadata description in RightField documents is extracted and stored in a knowledge graph, but as mentioned before it is not being actually used in any form by the current SEEK version. No batch or programmatic upload is possible.	There are two versions of OpenBIS interface: classic and a new via not officially released ELN-LIMS plugin. The classic interface allows metadata description and defining relationships between entities, it is usable, even if not particularly user friendly. However, the actual data file upload is totally impractical. File deposition is done through an applet, which has to be downloaded for each upload. Due to current JAVA security restrictions it involves accepting multiple warnings and it is a lengthy process. Also, due to local security policies it may not be permitted on some of the university's computers. ELN plugin interface is modern and more user friendly but less powerful than the classic one. However, its features are probably sufficient for the typical users. Unlike the classic one, it offers seamless and convenient option for data file upload. OpenBIS allows batch data descriptions. An excel table can be upload in which columns represents properties of the entities and rows individual entries. We successfully used this option to migrate data from an external DB to their OpenBIS representation. Programmatic data description and deposition is possible using custom scripts invoked by the dropbox mechanism. The scripts can extract metadata from the file content or determine desired logical structure of the data. This mechanism is extremely useful for automation of the data upload. There is also remote API for manipulation of the metadata and data in OpenBIS. We successfully used it for metadata updates, but we were unable to create new OpenBIS entities with this API.
Hooks for processing and systems integrations	
There are none available at the moment. There used to be a "harvester" mechanism for data indexing, but it has been discontinued and we were warned against using it.	There is an API for remote access to OpenBIS, it works well for read operations but lacks functionality for the write ones.

	There is a possibility to access data in OpenBIS using FTP protocol but we did not investigate this path. For write access (data deposit) external systems can utilize the dropbox mechanism as we did in the PS use case.
Developer experience	
The source code for SEEK is open source (BSD License) and is available on GitHub. The installation process is very well documented, although we did have to alter it slightly to install on the Scientific Linux OS available at SynthSys, rather than the recommended Ubuntu OS. There is a discussion forum for SEEK developers, and the core developers are responsive and helpful - https://groups.google.com/forum/#!forum/seek-developers The code is written in a standard Ruby on Rails way. It has many tests, which aid in the understanding of the code, and development of it. We added a feature to allow the SEEK admin to change the default access permissions, and this change was accepted into the mainstream SEEK code.	Although source code for OpenBIS is available online there is no description of how to build it. There is no information on how the code repository is organized with its multiple subprojects and versions. As a result it is not possible to build our own version of OpenBIS or introduce to it drastic changes (for example change the security model). OpenBIS seems to have a powerful plugin mechanism for developing extensions which can change how the data and metadata are handled. But it is a rich mechanism and not documented enough to be efficiently used by external developers. We treated OpenBIS as closed-software and used only the exposed API and dropbox mechanism to build necessary features around OpenBIS rather than integrate them within it.
Sustainability on the centre wide level	
We did not encounter any issues that could cause problems for centre wide usage.	Potential issues: • Security model that requires admin privileges to modify metadata model. Changes to closed vocabularies or experiment types should be done by a data manager on the research group level, instead of by admin on the centre level • Selection of options for entities types or attributes or interest (for search) are populated using the all existing properties. It may make it problematic to use once multiple, different types are defined for many, individual research groups. • There is no option for granting public access to already existing private resources

Costs of maintenance and customization

In this pilot project we were learning the systems in the roles of user, admin and developer. It is difficult to give an accurate estimate for the running costs as we were still in the discovery phase.

Both systems are currently under development and their regular update will be necessary. We estimate the time necessary for installation or update of each of them as one working day [revision: an OpenBIS installation that replicated a working system has failed to load the Jython functions necessary for data analysis, after a full day's troubleshooting of this feature alone].

Adding a new group that just wishes to use the basic functionality of the systems involves a small admin task and the necessary user training. Such preparation would take one day in SEEK and two to three days in OpenBIS as the latter is less intuitive and involves the definition of meta types.

However, effective data management systems need to offer added value to the users in order to encourage their daily usage. A software developer (programmer) who can extend their functionality to cater for particular research groups is essential.

Based on our experience with SEEK and the PS use case, automatic metadata extraction, its representation in RDF and a custom search page would take about two weeks to implement. Ideally, a generic RDF search facility could be added to SEEK but this is a larger task that would involve some research and could take 2-3 months. Similar effort would be necessary in order to incorporate data processing into SEEK. OpenBIS supported both functions for our PS use case.

In the case of OpenBIS we estimate the time necessary to implement a new use case similar to PS as three weeks of developer time. In that time it should be possible to import existing inventories into OpenBIS, implement metadata extraction and enable data processing.

Aggregating these tasks, we estimate that establishing data management support on these platforms for the SynthSys-Mammalian project with its 15 disparate research groups would require a full time developer for one year. After that initial phase of setting up, customization and user training, the workload could be reduced to 25% of FT for the developer. This should still support some new data types and functions, albeit more slowly. Both phases assume a minor proportion of server administration tasks, not including provision and maintenance of hardware and operating systems. Any data management also requires new roles for users, such as a Data Manager within research groups or projects and a Data Curator at the Centre or School level, which are beyond the scope of this technical evaluation.

Conclusion

We recommend that both OpenBIS and SEEK should be available to SynthSys as they provide useful and complementary functionality.

SEEK can easily perform the role of a collaboration platform with good provision for public access, profiting from a flexible permission model, ISA data organization and build-in support for publications and biological models. The strengths of OpenBIS lie in catering for fine-grained, structured metadata that can be automatically extracted from uploaded files, and in participation in automated data processing workflows. The potential of both systems will be greatly increased after their planned integration at the beginning of 2016 by the FAIRDOM consortium.

Our conclusions highlight the importance of the practical evaluation of the software. Naturally, both systems need additional, case-by-case, software development in order to provide the important "extra value" for the researcher. These specialised features will greatly facilitate their adoption and incorporation into daily research workflows. However, the simple use cases led us to an assessment of the systems' features that substantially differs from our initial expectations based upon their documentation. Likewise, our conclusions contrast with a recent literature overview by Wruck et al. 2014. The review praises SEEK for its support for rich metadata, RightField templates and automated data harvesting, exactly the elements that fell short in our PS use case. The same report stresses the importance of fine-grained access control but it does not comment on the very constrained, simplified model of OpenBIS, which eliminated it from our PlaSMo use case, nor on the problems with data upload using the OpenBIS web interface.

Literature

Wruck W, Peuker M, Regenbrecht CRA. Data management strategies for multinational large-scale systems biology projects. Briefings in Bioinformatics. 2014;15(1):65-78. doi:10.1093/bib/bbs064.